PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-335181
(43)Date of publication of application : 17.12.1996

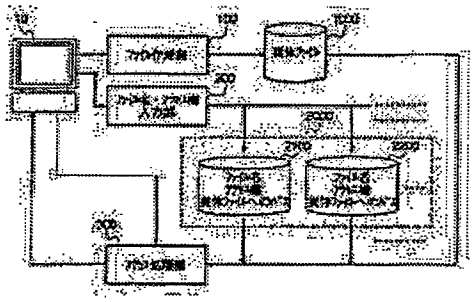
(51)Int.Cl. G06F 12/00
G06F 12/00

(21)Application number : 07-163025
(22)Date of filing : 06.06.1995

(71)Applicant : CANON INC
(72)Inventor : KATAYAMA HISASHI
IWABUCHI YOICHI
YABE TOSHIO

(54) FILE ACCESSING DEVICE

(57)Abstract:
PURPOSE: To control an access right by names by performing access processing to a file according to the access right corresponding to each name when performing access corresponding to that name.
CONSTITUTION: When the display command of a substance file (ORG file) 1000 is inputted, an access processing part 300 judges whether the name of the accessed file shows an access right storage part file (AS file) or not. When the name of the accessed file shows the AS file, it is judged whether the access name is (a) or (b). In the case of (a), the display command is executed according to the presence of a read right while using the access right stored in an AS file a2100. Therefore, the ORG file 1000 can be displayed. When the access file name is (b), on the other hand, the display command is executed without any read right while using the access right stored in an AS file b(2200). Therefore, the ORG file 1000 can not be splayed.



* NOTICES *

CLAIMS

[Claim(s)]
[Claim 1]It is an accessible file accessing device in the name of plurality about one file by an operating system characterized by comprising the following which manages a file system.
A memory measure which memorizes the right to access corresponding to said each name.
An access processing means which performs access processing to a file by the right to access corresponding to an each name based on a memory content of this memory measure.

[Translation done.]

DETAILED DESCRIPTION

[Detailed Description of the Invention]
[0001]
[Industrial Application]This invention relates one file to an accessible file accessing device in the name of plurality with the operating system which manages a file system.
[0002]
[Description of the Prior Art]Generally the conventional file accessing device can access one file by two or more names with the operating system which manages a file system. For example, in a UNIX file system, one entity file can be accessed in the name of plurality by stretching a Hard Link or Symbolic Link. The right to access is controlled by granting the right of direct access to the entity file.
[0003]
[Problem(s) to be Solved by the Invention]However, in the conventional example device, since the right to access to an entity file was controlled only by the entity file, when accessible in the name of plurality, the problem that the right to access was uncontrollable was according to the name about one file.
[0004]Then, this invention is made that the above-mentioned problem should be solved, and let it be the purpose of providing the file accessing device which can control the right to access for one file according to a name when accessible in the name of plurality.
[0005]
[Means for Solving the Problem]In order to attain an aforementioned problem, this invention is characterized by that an accessible file accessing device comprises the following in the name of plurality in one file with an operating system which manages a file system.
A memory measure which memorizes the right to access corresponding to said each name.
An access processing means which performs access processing to a file by the right to access corresponding to an each name based on a memory content of this memory measure.

[0006]
[Function]According to this invention of the above-mentioned composition, an access processing means performs access processing to a file by the right to access corresponding to the name, when accessing by an each name.
[0007]
[Example]Hereafter, working example of this invention is described in detail with reference to Drawings.
[0008]Drawing 1 is a lineblock diagram showing one working example of the file accessing device of this invention.
[0009]The input/output device 10 constituted so that this example device might be provided with the data output part which consists of a data input part which consists of keyboards etc., CRT, etc. as shown in the figure and edit of a file could be performed with an operator, The file creation part 100 which creates the entity file (henceforth "an ORG file") 1000 by the file creation command inputted from the input/output device 10, The file name and the right-to-access input part 200 which memorizes the right-to-access storage parts store (henceforth "the AS section") 2000, and the file name (alias of an ORG file) and the right to access which are inputted from the input/output device 10, and the path to the ORG file 1000 in the AS section 2000, It has the access processing part 300 which performs access processing to the ORG file 1000 based on the file name (alias of an ORG file) and the right to access which were memorized by the AS section 2000, and the path to the ORG file 1000, and is constituted.

[0010]Next, operation of this example is explained according to the flow chart of drawing 3 based on the example of drawing 2.
[0011]Drawing 2 is the list of the creation files for explaining an example. a is a file name (alias of an ORG file) of the file (henceforth "the AS file a") 2100 memorized by the AS section 2000 among the figure, b is a file name (alias of an ORG file) of the file (henceforth "the AS file b") 2200 memorized by the AS section 2000. In the case of this example, for the AS file a (2100). a which is an alias of the ORG file 1000, the right to access used when the ORG file 1000 is accessed by the name a, and the path which gets to know where the ORG file 1000 is when the ORG file 1000 is accessed by the name a are memorized. b which is an alias of the ORG file 1000 at the AS file b (2200), The right to access used when the ORG file 1000 is accessed by the name b, and the path which gets to know where the ORG file 1000 is when the ORG file 1000 is accessed by the name b are memorized. If the right to access of the AS files a and b is seen and it will access by the name a, it is shown that the ORG file 1000 can be accessed without the right of a lead with the right of a lead if it accesses by the name b. Therefore, the right to access of the ORG file 1000 is dependent on file name a and b. Although it does not especially become a problem by next explanation where the ORG file 1000 is, the path to the ORG file 1000 is once set to /usr/ORG.
[0012]First, an operator operates the input/output device 10 and creates the entity file (ORG file) 1000 (S1). Although the AS section 2000 may be created at the time of ORG file 1000 creation before creating the ORG file 1000, when an alias is attached to the ORG file 1000, by the following explanation, the case where (S2) and the AS section 2000 are also created is made into an example. Although the file creation part 100, and a file name and a right-to-access input part 200 are another, these may be performed by the same preparing part. The AS section 2000 may be memorized in the ORG file 1000.
[0013]Next, an operator creates the AS file a of file name a which operates the input/output device 10 and is equivalent to the AS section 2000 (2100), and the AS file b of file name b (2200) (S2).
[0014]The following composition is also considered by the preparing means of the AS section 2000 although the AS section 2000 is created as an AS file according to name in this step S2.
[0015]. Namely, memorize and manage the memory content of the AS section 2000 according to a name (ex.211). . Memorize and manage the memory content of the AS section 2000 in a database every ORG file 1000 (ex.212). It is using media (tape etc.) other than a file in memory of the AS section 2000 which uses a file (AS file) for memory of the AS section 2000 (ex.214) which manages the memory content of the AS section 2000 in a unified manner in one database (ex.213) (ex.215).
[0016]Although the right to access of the AS files a and b may not be determined at the time of AS section 2000 creation at Step S2, a system may be determined freely (ex.221) and may use the right to access of the ORG file 1000 (ex.222), and the right to access may be boiled, respectively at the time of creation of the AS files a and b, and may receive and grant it (ex.223).
[0017]It is a stage of this step S2, and the access by the name of ORG can forbid, or can also manage the name which ORG file 1000 self has, and the right to access in the AS section 2000. By drawing 2, when forbidding access to the name of ORG or also managing the name and the right to access of ORG in the AS section 2000, it shall not think.
[0018]In Step S3, a file name and the right-to-access input part 200 store the right to access corresponding to an each name in the AS section 2000. Here, the right to access of the AS file a (2100) adds the right of a lead, and as the right to access of the AS file b (2200) does not add the right of a lead, it is memorized by the AS section 2000.
[0019]In the case of the example of drawing 2, although carrying out also at Step S2 cuts Step S3, it is made into another step in consideration of a user changing the right to access afterwards here (Step S3 can be performed at any time, if the AS section 2000 is created).
[0020]In drawing 3, since the flow chart is simplified, the user is omitting the portion which changes the right to access repeatedly if needed.
[0021]Although only the right of a lead was taken for the example as the right to access, if it is the right which is controlling access to a file, what kind of the right to access may be controlled by drawing 2 besides the right of a lead. Some examples of the right to access in a UNIX file system are given to below.
[0022]Namely, the right of a lead to the person himself/herself (ex.331), the light right to the person himself/herself (ex.332), There are the execution right (ex.333) to the person himself/herself, an execution right (ex.334) to others, a right of a lead to a group (ex.335), a light right (ex.336) to a group, an execution right (ex.337) to a group, etc.

[0023]At step S4, the access processing part 300 accesses the entity file 1000 by the right to access according to name based on the information stored in the AS section 2000. In the case of drawing 2, if it is going to lead the ORG file 1000 by the name a, it can lead, but even if it is going to lead the ORG file 1000 by the name b, it cannot lead.
[0024]When accessing the ORG file 1000 by step S4, the method of the following access control can be considered.
[0025]Namely, when accessing the ORG file 1000 by the name stored in the AS section 2000, . The right to access which the ORG file 1000 has is disregarded, and validate only the right to access stored in the AS section 2000 (ex.411). When accessing the ORG file 1000 by the name stored in the AS section 2000, By changing the right to access of the ORG file 1000 into the right to access stored in the AS section 2000. It is controlling the right to access according to a name (ex.412) (both of also returning the right to access of the ORG file 1000 before right-to-access change after the access processing in a or b and things which are not returned are possible) etc.
[0026]Drawing 4 is the example which detailed Step S2 of drawing 3 using the example of a preparing means of the AB section 2000 (ex.211,214,222).
[0027]An operator operates the input/output device 10 and attaches the alias a to the ORG file 1000 (S21). At this time, a file name and the right input part 200 of an accelerator memorize the path to the file name and the ORG file 1000 a, to the AS file a (2100). Next, the right to access of the ORG file 1000 is memorized to the AS file a (2100) (S22). Then, an operator operates the input/output device 10 and attaches the alias b to the ORG file 1000 (S23). At this time, a file name and the right input part 200 of an accelerator memorize the path to the file name and the ORG file b, to the AS file b (2200). Next, the right to access of the ORG file 1000 is memorized to the AS file b (2200) (S24).
[0028]The point of the turn which may perform simultaneously said steps S21 and S22 and said SUTEPU S23 and S24, and creates the alias a and b may be sufficient as whichever here.
[0029]Drawing 5 is the example which detailed Step S2 of drawing 3 using the example of a preparing means of the AB section 2000 (ex.212,213,222).
[0030]An operator operates the input/output device 10 and carries out an alias (a or b) input to the ORG file 1000 (S25). The existence of an access right table (AS section) is investigated (S26), and if there are not a file name and the right input part 200 of an accelerator, an access right table (AS section) is created (S27), and if it is, it will be registered into an access right table (AS section) (S28).
[0031]Drawing 6 is the example which Step S2 used step S4 of drawing 3, and drawing 4 and an access control gestalt used (ex.411), and was detailed.
[0032]The access processing part 300 will judge whether the accessed file name (accessed file name) is AS file, if the display commands of the ORG file 1000 enter (S411) (S412). When an access file name is AS file, An access name judges a or b (S413), and since display commands will be executed with the right of a lead using the right to access stored in the AS file a (2100) if it is a (S415), the ORG file 1000 can be displayed. In said step S413, since display commands are executed without the right of a lead using the right to access stored in the AS file b (2200) when an access file name is b (S416), the ORG file 1000 cannot be displayed.
[0033]the case where an access file name is not AS file in said step S412 — (— the file name in this case executes display commands by the right to access which ORG) and the ORG file 1000 have then (S414).
[0034]Drawing 7 is the example which Step S2 used step S4 of drawing 3, and drawing 4 and an access control gestalt used (ex.412), and was detailed.
[0035]The access processing part 300 will judge whether an access file name is AS file, if the display commands of the ORG file 1000 enter (S421) (S422). When an access file name is AS file, An access file name judges a or b (S413), and when a file name is a, Since the right to access of the ORG file 1000 is changed into the right to access stored in the AS file a (2100) (S424) and the display commands of the ORG file 1000 are executed (S426), the ORG file 1000 can be displayed. In said step S423, when an access file name is b, Although the right to access of the ORG file 1000 is changed into the right to access stored in the AS file b (2200) (S425) and the display commands of the ORG file 1000 are executed (S426), the ORG file 1000 cannot be displayed. If a file name is not AS file at said step S422 (the file name in this case is ORG), the ORG file 1000 will execute display commands by the right to access which it has then.
[0036]As explained above, when the operating system which operates on a computer accesses one file by two or more names according to this example, the ORG file 1000 can be accessed based on the access right memorized according to the name. Thereby, the effect that access to a file can be performed by the right to

access corresponding to each name is acquired to two or more names (alias) one file was named.
[0037]
[Effect of the Invention]In this invention explained in full detail above, when memorizing the right to access corresponding to an each name and accessing by an each name, the right to access corresponding to the name performs access processing to a file.
Therefore, the file accessing device which can control the right to access for one file according to a name when accessible in the name of plurality can be provided.

[Translation done.]

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]
[Drawing 1]It is a lineblock diagram of this example.
[Drawing 2]It is a figure showing the list of the creation files of this example.
[Drawing 3]It is a flow chart which shows operation of this example.
[Drawing 4]It is a flow chart which shows the example (ex.211,214,222 is used) which detailed Step S2 of drawing 3.
[Drawing 5]It is a flow chart which shows the example (ex.212,213,222 is used) which detailed Step S2 of drawing 3.
[Drawing 6]It is a flow chart which shows the example (S2 uses drawing 4 and an access control gestalt uses (ex.411)) which detailed step S4 of drawing 3.
[Drawing 7]It is a flow chart which shows the example (S2 uses drawing 4 and an access control gestalt uses (ex.412)) which detailed step S4 of drawing 3.
[Description of Notations]
10 Input/output device
100 File creation part
200 A file name and a right-to-access input part
300 Access processing part
1000 Entity file (ORG file)
2000 Right-to-access storage parts store (AS section)
2100-2200 AS file

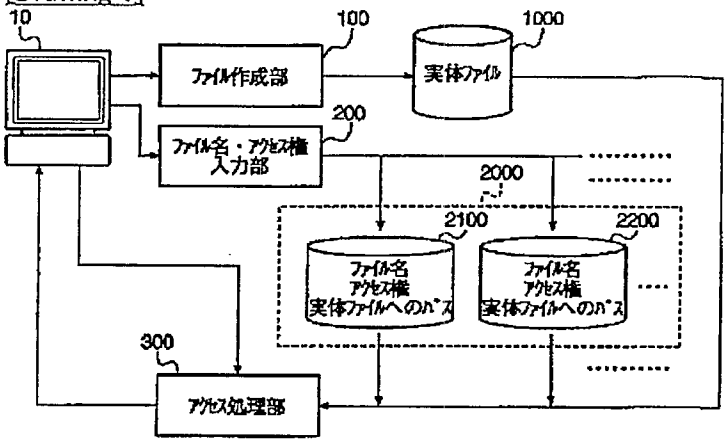
[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DRAWINGS

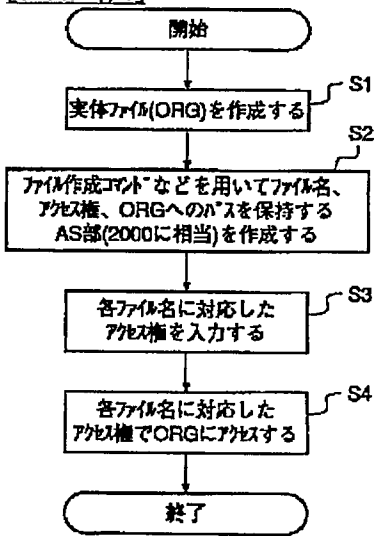
[Drawing 1]



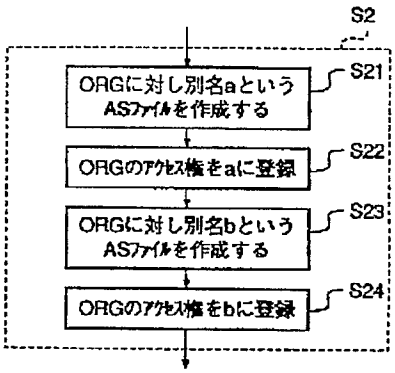
[Drawing 2]

ファイル名	a	b	実体ファイルORG
アクセス権	リード権有	リード権無	呼び出される名前に依存
実体ファイルへのパス	/usr/ORG	/usr/ORG	.....

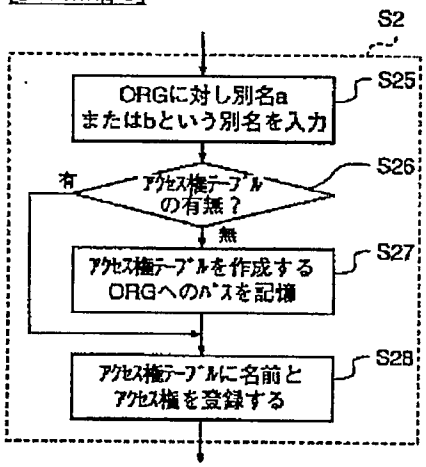
[Drawing 3]



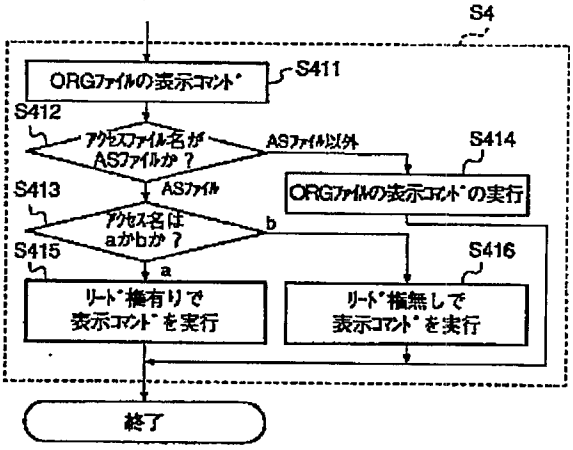
[Drawing 4]



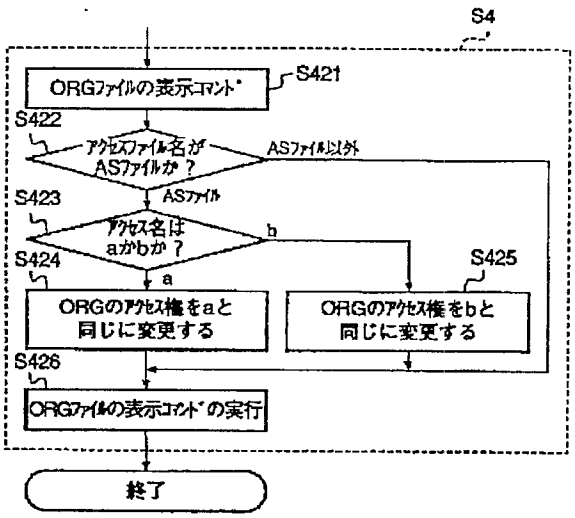
[Drawing 5]



[Drawing 6]



[Drawing 7]



[Translation done.]

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（A）　(11)特許出願公開番号

特開平8－335181

(43)公開日　平成８年(1996)12月17日

(51)Int.Cl.[6]	識別記号	庁内整理番号	ＦＩ		技術表示箇所
G06F 12/00	537	7623-5B	G06F 12/00	537	A
	514	7623-5B		514	R

審査請求　未請求　請求項の数１　ＦＤ　（全６頁）

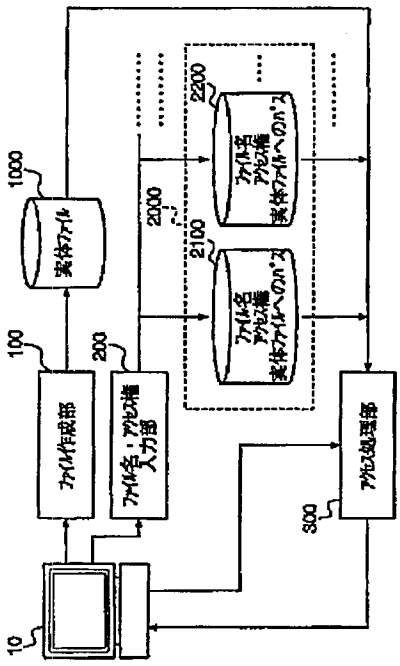
(21)出願番号　特願平7－163025

(22)出願日　平成７年(1995)６月６日

(71)出願人　000001007
キヤノン株式会社
東京都大田区下丸子３丁目30番２号
(72)発明者　片山　尚志
東京都大田区下丸子３丁目30番２号　キヤノン株式会社内
(72)発明者　岩渕　洋一
東京都大田区下丸子３丁目30番２号　キヤノン株式会社内
(72)発明者　矢部　俊夫
東京都大田区下丸子３丁目30番２号　キヤノン株式会社内
(74)代理人　弁理士　渡部　敏彦

(54)【発明の名称】ファイルアクセス装置

(57)【要約】
【目的】　一つのファイルを複数の名前でアクセス可能である場合に、名前別にアクセス権を制御することが可能なファイルアクセス装置を提供する。
【構成】　アクセル処理部３００は、各名前でアクセスする場合に、その名前に対応するアクセス権でファイルへのアクセス処理を行う。



1

【特許請求の範囲】

【請求項１】　ファイルシステムを管理するオペレーティングシステムにより１つのファイルを複数の名前でアクセス可能なファイルアクセス装置において、前記各名前に対応してアクセス権を記憶する記憶手段と、この記憶手段の記憶内容を基に各名前に対応するアクセス権でファイルへのアクセス処理を行うアクセス処理手段とを有することを特徴とするファイルアクセス装置。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】本発明は、ファイルシステムを管理するオペレーティングシステムにより一つのファイルを複数の名前でアクセス可能なファイルアクセス装置に関する。

【０００２】

【従来の技術】従来のファイルアクセス装置は、一般に、ファイルシステムを管理するオペレーティングシステムにより一つのファイルを複数の名前によってアクセスすることが可能となっている。例えば、ＵＮＩＸファイルシステムでは、ハードリンクまたはシンボリックリンクを張ることによって、一つの実体ファイルを複数の名前でアクセスすることができる。さらに、その実体ファイルに直接アクセス権を与えることによって、アクセス権の制御を行っている。

【０００３】

【発明が解決しようとする課題】しかしながら、従来例装置では、実体ファイルへのアクセス権を実体ファイルでのみ制御しているため、一つのファイルを複数の名前でアクセス可能である場合に、名前別にアクセス権を制御することができないという問題があった。

【０００４】そこで、本発明は、上記問題点を解決すべくなされたものであり、一つのファイルを複数の名前でアクセス可能である場合に、名前別にアクセス権を制御することが可能なファイルアクセス装置を提供する目的とするものである。

【０００５】

【課題を解決するための手段】上記課題を達成するために本発明は、ファイルシステムを管理するオペレーティングシステムにより１つのファイルを複数の名前でアクセス可能なファイルアクセス装置において、前記各名前に対応してアクセス権を記憶する記憶手段と、この記憶手段の記憶内容を基に各名前に対応するアクセス権でファイルへのアクセス処理を行うアクセス処理手段とを有することを特徴とするものである。

【０００６】

【作用】上記構成の本発明によれば、アクセス処理手段は、各名前でアクセスする場合に、その名前に対応するアクセス権でファイルへのアクセス処理を行う。

【０００７】

【実施例】以下、本発明の実施例を図面を参照して詳細に説明する。

2

【０００８】図１は本発明のファイルアクセス装置の一実施例を示す構成図である。

【０００９】本実施例装置は、同図に示すように、キーボード等からなるデータ入力部とＣＲＴ等からなるデータ出力部とを備え、オペレータによりファイルの編集ができるように構成された入出力装置１０と、入出力装置１０から入力されるファイル作成コマンドにより実体ファイル（以下「ＯＲＧファイル」ともいう。）１０００を作成するファイル作成部１００と、アクセス権記憶部（以下「ＡＳ部」という。）２０００と、入出力装置１０から入力されるファイル名（ＯＲＧファイルの別名）とアクセス権とＯＲＧファイル１０００へのパスとをＡＳ部２０００に記憶するファイル名・アクセス権入力部２００と、ＡＳ部２０００に記憶されたファイル名（ＯＲＧファイルの別名）とアクセス権とＯＲＧファイル１０００へのパスとを基にＯＲＧファイル１０００へのアクセス処理を行うアクセス処理部３００とを有して構成されている。

【００１０】次に、本実施例の動作を図２の具体例を基に図３のフローチャートに従って説明する。

【００１１】図２は具体例を説明するための作成ファイルの一覧である。同図中、ａはＡＳ部２０００に記憶されるファイル（以下「ＡＳファイルａ」という。）２１００のファイル名（ＯＲＧファイルの別名）であり、ｂはＡＳ部２０００に記憶されるファイル（以下「ＡＳファイルｂ」という。）２２００のファイル名（ＯＲＧファイルの別名）である。この例の場合は、ＡＳファイルａ（２１００）には、ＯＲＧファイル１０００の別名であるａと、ＯＲＧファイル１０００がａという名前でアクセスされたときに使用されるアクセス権と、ＯＲＧファイル１０００がａという名前でアクセスされたときにＯＲＧファイル１０００がどこにあるのかを知るパスとが記憶される。また、ＡＳファイルｂ（２２００）には、ＯＲＧファイル１０００の別名であるｂと、ＯＲＧファイル１０００がｂという名前でアクセスされたときに使用されるアクセス権と、ＯＲＧファイル１０００がｂという名前でアクセスされたときにＯＲＧファイル１０００がどこにあるのかを知るパスとが記憶される。また、ＡＳファイルａ，ｂのアクセス権を見れば、ａという名前でアクセスすればリード権ありで、ｂという名前でアクセスすればリード権なしでＯＲＧファイル１０００にアクセスすることができることを示している。従って、ＯＲＧファイル１０００のアクセス権は、ファイル名ａ，ｂに依存している。ＯＲＧファイル１０００がどこにあるかは、この後の説明で特に問題になることはないが、一応ＯＲＧファイル１０００へのパスは／ｕｓｒ／ＯＲＧとしておく。

【００１２】まず、オペレータは、入出力装置１０を操作して実体ファイル（ＯＲＧファイル）１０００を作成

する（Ｓ１）。なお、ＯＲＧファイル１０００を作成する前か、あるいはＯＲＧファイル１０００作成時に、ＡＳ部２０００を作成してもよいが、以下の説明では、ＯＲＧファイル１０００に対して別名をつける時に（Ｓ２）、ＡＳ部２０００も作成する場合を例にする。また、ファイル作成部１００と、ファイル名・アクセス権入力部２００とが別になっているが、これらは同一作成部で行ってもかまわない。また、ＯＲＧファイル１０００の中でＡＳ部２０００を記憶してもかまわない。

【００１３】次に、オペレータは、入出力装置１０を操作してＡＳ部２０００に相当するファイル名ａのＡＳファイルａ（２１００）及びファイル名ｂのＡＳファイルｂ（２２００）を作成する（Ｓ２）。

【００１４】なお、このステップＳ２では、ＡＳ部２０００を名前別のＡＳファイルとして作成することにしているが、ＡＳ部２０００の作成手段には、以下のような構成も考えられる。

【００１５】すなわち、ＡＳ部２０００の記憶内容を名前別に記憶し管理する（ｅｘ．２１１）、ＡＳ部２０００の記憶内容をＯＲＧファイル１０００ごとにデータベースに記憶し管理する（ｅｘ．２１２）、ＡＳ部２０００の記憶内容を一つのデータベースで一元管理する（ｅｘ．２１３）、ＡＳ部２０００の記憶にはファイル（ＡＳファイル）を使用する（ｅｘ．２１４）、ＡＳ部２０００の記憶にはファイル以外の媒体（テープ等）を使用する（ｅｘ．２１５）等である。

【００１６】また、ステップＳ２でのＡＳ部２０００作成時には、ＡＳファイルａ，ｂのアクセス権が決定されていない場合があるが、そのアクセス権は、システムが自由に決定してもよく（ｅｘ．２２１）、ＯＲＧファイル１０００のアクセス権を使用してもよく（ｅｘ．２２２）、ＡＳファイルａ，ｂの作成時にそれぞれに対して与えてもよい（ｅｘ．２２３）。

【００１７】さらに、このステップＳ２の段階で、ＯＲＧという名前でのアクセスは禁止したり、ＯＲＧファイル１０００自身が持つ名前とアクセス権もＡＳ部２０００で管理することも可能である。図２ではＯＲＧという名前に対してアクセスを禁止したり、ＯＲＧの名前とアクセス権もＡＳ部２０００で管理する場合は考えないものとする。

【００１８】ステップＳ３では、ファイル名・アクセス権入力部２００は、各名前に対応したアクセス権をＡＳ部２０００に格納する。ここではＡＳファイルａ（２１００）のアクセス権はリード権を付加し、ＡＳファイルｂ（２２００）のアクセス権はリード権を付加しないようにしてＡＳ部２０００に記憶される。

【００１９】なお、図２の例の場合は、ステップＳ３はステップＳ２でも行うことがきるが、ここではユーザが後からアクセス権を変更することを考慮して、別ステップとする（ステップＳ３はＡＳ部２０００が作成されていればいつでも行うことができる）。

【００２０】また、図３では、フローチャートを簡略化しているため、ユーザが必要に応じて何度もアクセス権を変更する部分は省略している。

【００２１】さらに、図２では、アクセス権としてリード権のみを例にとったが、ファイルへのアクセスを制御している権利ならリード権以外にもどんなアクセス権を制御してもかまわない。以下にＵＮＩＸファイルシステムでのアクセス権の例をいくつか挙げる。

【００２２】すなわち、本人に対するリード権（ｅｘ．３３１）、本人に対するライト権（ｅｘ．３３２）、本人に対する実行権（ｅｘ．３３３）、他人に対する実行権（ｅｘ．３３４）、グループに対するリード権（ｅｘ．３３５）、グループに対するライト権（ｅｘ．３３６）、グループに対する実行権（ｅｘ．３３７）等がある。

【００２３】ステップＳ４では、アクセス処理部３００が、ＡＳ部２０００に格納されている情報を基に名前別のアクセス権で実体ファイル１０００にアクセスする。図２の場合、ａという名前でＯＲＧファイル１０００をリードしようとすればリードできるが、ｂという名前でＯＲＧファイル１０００をリードしようとしてもリードできない。

【００２４】また、ステップＳ４でＯＲＧファイル１０００にアクセスする場合に、以下のようなアクセス制御の仕方が考えられる。

【００２５】すなわち、ＡＳ部２０００に格納されている名前でＯＲＧファイル１０００にアクセスする場合は、ＯＲＧファイル１０００が持っているアクセス権を無視して、ＡＳ部２０００に格納されているアクセス権のみを有効にする（ｅｘ．４１１）、ＡＳ部２０００に格納されている名前でＯＲＧファイル１０００にアクセスする場合は、ＯＲＧファイル１０００のアクセス権をＡＳ部２０００に格納されているアクセス権に変更することで、名前別にアクセス権の制御を行う（ａまたはｂでのアクセス処理後、ＯＲＧファイル１０００のアクセス権をアクセス権変更前に戻すことも戻さないこともどちらも可能である）（ｅｘ．４１２）等である。

【００２６】図４は、図３のステップＳ２を、ＡＢ部２０００の作成手段例（ｅｘ．２１１，２１４，２２２）を用いて詳細化した例である。

【００２７】オペレータは、入出力装置１０を操作してＯＲＧファイル１０００に対して別名ａをつける（Ｓ２１）。この時、ファイル名・アクセル権入力部２００は、ａというファイル名とＯＲＧファイル１０００へのパスをＡＳファイルａ（２１００）に記憶する。次に、ＯＲＧファイル１０００のアクセス権をＡＳファイルａ（２１００）に記憶する（Ｓ２２）。続いて、オペレータは、入出力装置１０を操作してＯＲＧファイル１０００に対して別名ｂをつける（Ｓ２３）。この時、ファイ

ル名・アクセル権入力部２００は、ｂというファイル名とＯＲＧファイルへのパスをＡＳファイルｂ（２２００）に記憶する。次に、ＯＲＧファイル１０００のアクセス権をＡＳファイルｂ（２２００）に記憶する（Ｓ２４）。

【００２８】ここで前記ステップＳ２１とＳ２２、前記ステプＳ２３とＳ２４は、同時に行ってもよく、また、ａ，ｂという別名を作成する順番はどちらが先でもよい。

【００２９】図５は、図３のステップＳ２を、ＡＢ部２０００の作成手段例（ｅｘ．２１２，２１３，２２２）を用いて詳細化した例である。

【００３０】オペレータは、入出力装置１０を操作してＯＲＧファイル１０００に対し別名（ａまたはｂ）入力する（Ｓ２５）。ファイル名・アクセル権入力部２００は、アクセス権テーブル（ＡＳ部）の有無を調べ（Ｓ２６）、なければアクセス権テーブル（ＡＳ部）を作成し（Ｓ２７）、有ればアクセス権テーブル（ＡＳ部）に登録する（Ｓ２８）。

【００３１】図６は、図３のステップＳ４を、ステップＳ２は図４、アクセス制御形態は（ｅｘ．４１１）を用いて詳細化した例である。

【００３２】アクセス処理部３００は、ＯＲＧファイル１０００の表示コマンドが入って来ると（Ｓ４１１）、アクセスしたファイル名（アクセスしたファイル名）がＡＳファイルか否かを判断する（Ｓ４１２）。アクセスファイル名がＡＳファイルである場合は、アクセス名がａかｂかを判断し（Ｓ４１３）、ａであれば、ＡＳファイルａ（２１００）に格納されているアクセス権を用いて、リード権有りで表示コマンドを実行するので（Ｓ４１５）、ＯＲＧファイル１０００を表示できる。前記ステップＳ４１３において、アクセスファイル名がｂである場合は、ＡＳファイルｂ（２２００）に格納されているアクセス権を用いて、リード権無しで表示コマンドを実行するので（Ｓ４１６）、ＯＲＧファイル１０００を表示できない。

【００３３】前記ステップＳ４１２において、アクセスファイル名がＡＳファイルでない場合は（この場合のファイル名はＯＲＧ）、ＯＲＧファイル１０００がその時持っているアクセス権で表示コマンドを実行する（Ｓ４１４）。

【００３４】図７は、図３のステップＳ４を、ステップＳ２は図４、アクセス制御形態は（ｅｘ．４１２）を用いて詳細化した例である。

【００３５】アクセス処理部３００は、ＯＲＧファイル１０００の表示コマンドが入って来ると（Ｓ４２１）、アクセスファイル名がＡＳファイルか否かを判断する（Ｓ４２２）。アクセスファイル名がＡＳファイルである場合は、アクセスファイル名がａかｂかを判断し（Ｓ４１３）、ファイル名がａである場合は、ＯＲＧファイル１０００のアクセス権をＡＳファイルａ（２１００）に格納されているアクセス権に変更し（Ｓ４２４）、ＯＲＧファイル１０００の表示コマンドを実行するので（Ｓ４２６）、ＯＲＧファイル１０００を表示できる。前記ステップＳ４２３において、アクセスファイル名がｂである場合は、ＯＲＧファイル１０００のアクセス権をＡＳファイルｂ（２２００）に格納されているアクセス権に変更して（Ｓ４２５）、ＯＲＧファイル１０００の表示コマンドを実行するが（Ｓ４２６）、ＯＲＧファイル１０００を表示できない。前記ステップＳ４２２でファイル名がＡＳファイルでなければ（この場合のファイル名はＯＲＧ）、ＯＲＧファイル１０００がその時持っているアクセス権で表示コマンドを実行する。

【００３６】以上説明したように、本実施例によれば、コンピュータ上で動作するオペレーティングシステムが１つのファイルを２つ以上の名前でアクセスした場合に、名前別に記憶されたアクセス権を基にＯＲＧファイル１０００にアクセスをすることができる。これにより、１つのファイルに付けられた複数の名前（別名）に対して、それぞれの名前に対応したアクセス権でファイルへのアクセスを行うことができるという効果が得られる。

【００３７】

【発明の効果】以上詳述した本発明によれば、各名前に対応してアクセス権を記憶し、各名前でアクセスする場合に、その名前に対応するアクセス権でファイルへのアクセス処理を行うので、一つのファイルを複数の名前でアクセス可能である場合に、名前別にアクセス権を制御することが可能なファイルアクセス装置を提供することができる。

【図面の簡単な説明】

【図１】本実施例の構成図である。

【図２】本実施例の作成ファイルの一覧を示す図である。

【図３】本実施例の動作を示すフローチャートである。

【図４】図３のステップＳ２を詳細化した例（ｅｘ．２１１，２１４，２２２を使用）を示すフローチャートである。

【図５】図３のステップＳ２を詳細化した例（ｅｘ．２１２，２１３，２２２を使用）を示すフローチャートである。

【図６】図３のステップＳ４を詳細化した例（Ｓ２は図４、アクセス制御形態は（ｅｘ．４１１）を使用）を示すフローチャートである。

【図７】図３のステップＳ４を詳細化した例（Ｓ２は図４、アクセス制御形態は（ｅｘ．４１２）を使用）を示すフローチャートである。

【符号の説明】

１０　入出力装置

１００　ファイル作成部

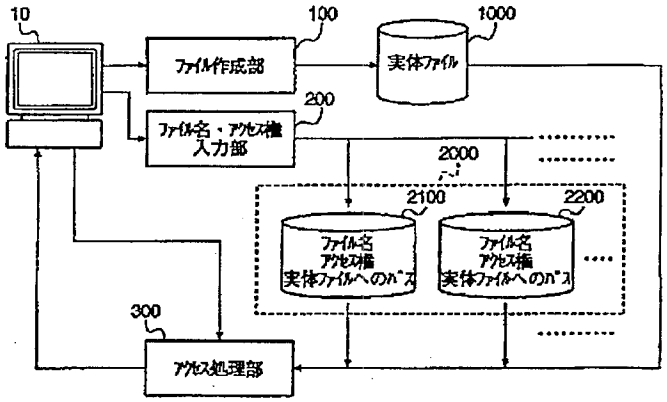
7

２００　ファイル名・アクセス権入力部
３００　アクセス処理部
１０００　実体ファイル（ＯＲＧファイル）

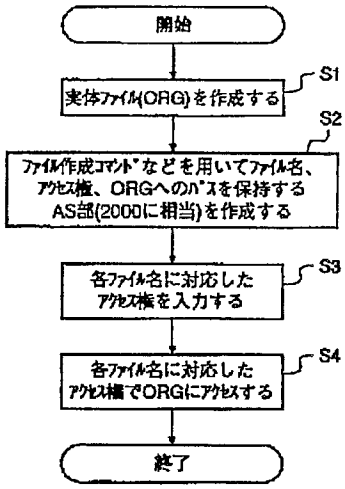
8

２０００　アクセス権記憶部（ＡＳ部）
２１００，２２００　ＡＳファイル

【図１】



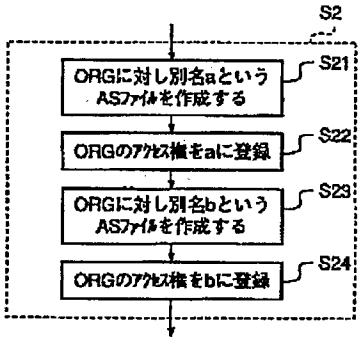
【図３】



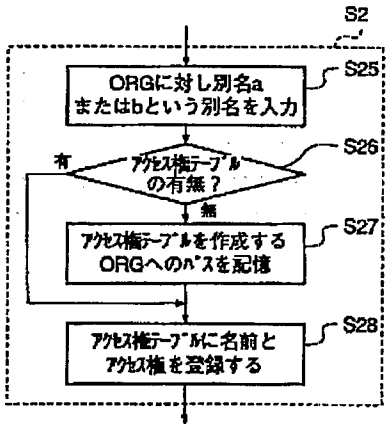
【図２】

ファイル名	a	b	実体ファイルORG
アクセス権	リード権有	リード権無	呼び出される名前に依存
実体ファイルへのパス	/usr/ORG	/usr/ORG	……

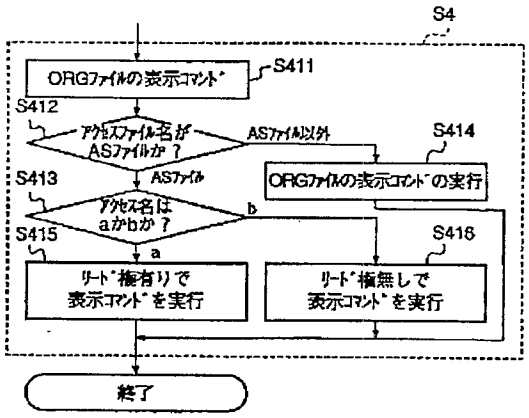
【図４】



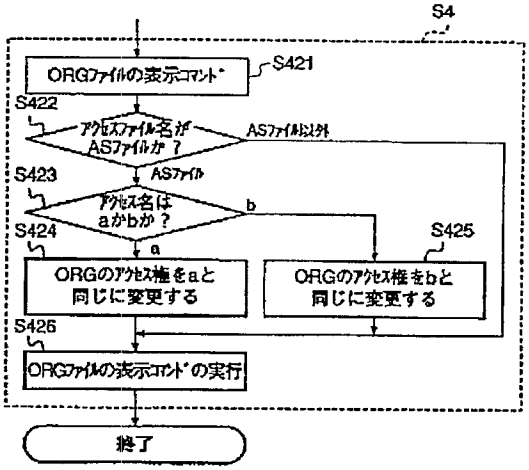
【図５】



【図６】


S4
ORGファイルの表示コマンド
S411
S412
アクセスファイル名がASファイルか？
ASファイル以外
S414
ORGファイルの表示コマンドの実行
S413
ASファイル
アクセス名はaかbか？
b
S415
a
S416
リード権有りで表示コマンドを実行
リード権無しで表示コマンドを実行
終了

【図７】


S4
ORGファイルの表示コマンド
S421
S422
アクセスファイル名がASファイルか？
ASファイル以外
S423
ASファイル
アクセス名はaかbか？
b
S424
a
S425
ORGのアクセス権をaと同じに変更する
ORGのアクセス権をbと同じに変更する
S426
ORGファイルの表示コマンドの実行
終了

Show

CONCEPT PROCEDURES

Importing CAD Files - Concepts

AGi32 allows for the input of CAD drawing files in DWG and DXF formats. There is no difference in the contents of the files, only the data type is different. DXFs are ASCII files, made of alphanumeric and numeric values describing the drawing file. As such, they can be quite large in size. DWG files are binary format files which makes them smaller. Regardless of the imported file format, all drawing, text and surface entities selected for import into AGi32 are formatted the same way in AGi32's internal database structure. Consequently, the AGi32 file size will be the same regardless of the original file format.

The DWG format is owned by Autodesk. AGi32 can read DWG files saved in Autodesk software versions 2008 and earlier.

Layer Selection

The following layer selection methods are available in both the Import CAD File-Layer Selection tab and the Import Mapping tab:

- Singular clicking or clicking and dragging to select one layer or one selection window of layers at one time.
- Shift-select - select adjacent layers and all selection in between top and bottom selections
- Ctrl-select - select non adjacent layers or non adjacent selection windows at once.

Imported Entity Types

AGi32 will import the following CAD created entities:

Entity	Comments
Supported Drawing Entity Types: ■ Line ■ Arc ■ Polyline (LwPolyline, 3DPolyline) ■ Spline ■ Circle ■ Ellipse (Elliptical Arc) ■ Text ■ Mtext (unformatted) ■ Mline ■ Trace ■ Solid (2D) ■ Dimensions (most types) ■ Leader (most types)	Drawing entity types are never imported as objects. Some drawing entities, such as those with thickness assigned, may look like 3D objects in CAD (they may even shade in CAD) but they are still 2D and will not be converted to objects in AGi32. Lines, Arcs, LWPolylines, Circles, Trace and 2D Solids with thickness assigned will be imported with defined thickness into AGi32. These are wireframe drawing entities and will not be converted into objects (surfaces) in AGi32. Most Mtext formatting is applied, however not all formatting unicode is interpreted. Curved entities (e.g. arcs, circles, etc.) will be reconstructed in AGi32 using the specified curve increment. It is recommended that you change the Curve Increment for Import to 5 degrees for exterior background drawings (as these environments typically have shallow curvature lines).
Supported Object Types: ■ 3D Face ■ Polyface Mesh ■ Polygon Mesh ■ Regions (holes within are supported) ■ 3D Solids (Planar surfaces only)	All zero area and degenerate (self intersecting) surfaces are not imported. All 3D Faces, Polyface and Polygon Meshes are imported. Non-planar and self intersecting surfaces will be deconstructed into two planar triangles. 3DSolids composed of planar surfaces (regions) are imported. Surfaces composed of NURBS surfaces or non-planar surfaces (bodies) will not be imported. Regions are imported including those with openings and curved edges created with splines, arcs and lines. Curved edges will be reconstructed in AGi32 using specified curve increment. Bodies may be converted to a Polyface Mesh in AutoCAD for importing into AGi32. See this AGi32 Forum Topic for details.
Other Supported Entities (Misc) ■ Block References (Inserts) ■ External Reference (XREF)	
Unsupported Entity Types:	

■ Attribute ■ Attribute Definitions ■ Body (3D) ■ Hatch ■ Hyperlink ■ MInsert Block ■ Point ■ Raster Image ■ Ray ■ Shape ■ Tolerance ■ XLine ■ Proxy ■ Paper Space (all entities in Paper Space)	More on bodies below (3D Entities section)

Units and Scaling

Units

AGi32 will read the AutoCAD units setting from the CAD file and will recognize either the "Engineering" or "Architectural" units settings (both are in inches). All other AutoCAD units settings (decimal, scientific etc.) are assumed to be the same as the current AGi32 units (i.e., if the current AGi32 units are Feet, the imported CAD file is also assumed to be in Feet; if the current AGi32 units are Meters, the imported CAD file is also assumed to be in Meters). In this case it is up to the user to select the correct units, if this assumption is incorrect.

Scaling

An additional Scaling Factor may be applied upon import in addition to any unit conversion scaling. The scaling factor may entered as a decimal value (e.g. 100, 0.75, etc.) or as a fraction (1/2, 7/8, etc.). Math operations are not supported (e.g., +, -, *). Scaling is generally recommended to correct any drawing extents abnormalities.

Drawing Extents

The Drawing Extents and drawing location are shown in this section. Drawing Extents displays the minimum extents (LLHC - Lower Left Hand Corner), maximum extents (URHC - Upper Right Hand Corner) and total extents in the selected AGi32 units.

Layers , Blocks and XREFs

AGi32 imports entities, both drawing and object, on a one to one basis. By default, each discreet entity becomes an individual AGi32 entity (either drawing entity or object). Entities are not grouped together into a single object unless specified by the user.

Grouping Drawing Entities

Drawing Entities may be grouped into a single layer in the Miscellaneous section of the Import CAD File dialog. The new Layer Name is specified by the user at this time. Before combining layers, it is important to consider how the CAD file will be used in the job file. Combining the entire file on one layer makes it easy to turn on or off or delete the CAD file, however, it may be desirable to maintain the layers and turn on or off or delete individual layers.

When drawing entities are grouped, all the drawing entities in all selected layers will be combined into one single layer. Drawing entities will retain their individual properties (e.g., color, text formatting, etc.).

Grouping Objects

3D entities may be grouped into a single object in the 3D Import Options section of the Import CAD File dialog. The new Object Name is specified by the user at this time. Each layer is made into an Object Component which may be assigned surface attributes as one definitive grouping. This method of grouping the layers together is only appropriate for entity import (where the imported file represents a single complex entity such as a plant, person, furniture, etc.).

When a whole 3D environment is imported in AGi32, grouping 3D Entities into a single object is strongly discouraged. Complex environments grouped together can cause coplanar surfaces (such as end panels within exterior walls) which may cause calculation errors and abrupt program termination in AGi32.

When 3D entities are grouped, all the objects in all selected layers will take on the drawing entity properties of the first layer and the surface attributes assigned by default, unless surface mapping.

When not grouped together, 3D entity file comes in as 2841 objects. Each object is selectable individually.

When grouped together, 3D entity file comes in as one object and edited, copied, rotated, etc. as one object.

Layer Selection

AGi32 provides four types of layer selection:

1. Smart Select Layers - This is the default layer selection method. This method only selects layers that are defined as visible (On and Unfrozen).
2. Individual Layer Selection - Layers may be selected individually through toggle selection. AGi32 support the Windows Shift and Ctrl methods for multiple layer selection.
3. Select All Layers - This option selects all layers
4. Clear Selected Layers - This option deselects all layers.

The Layers selection table only allows for layer selection, it does not provide property control (in other words you cannot switch a layer that is Off to On). These changes may only be applied in a CAD program prior to import in AGi32.

Layers that are selected for import are imported regardless of their properties. If layer selection methods other than Smart Select are employed, the AGi32 file may contain "empty" layers (layers that contain no imported data).

Blocks and XREFs

AGi32 does not currently support nested blocks (blocks within blocks), but only a single level of block 'hierarchy'. In the Blocks section, the user may choose to:

1. Explode all blocks on import - This default option destroys block hierarchy and uses layer hierarchy instead.
2. Keep Low-Level Blocks (only) - This option keeps only child blocks (blocks that do not contain any other blocks) intact. All blocks that contain other blocks are exploded. Each exploded entity assumes its properties (color, etc.) based on the layer in which it resides.
3. Keep High-Level Blocks (only) - This option keeps only parent blocks (blocks that are not contained within any other block) intact. All child blocks contained within the parent block are imported as part of and assume all properties (color, etc.) of the parent block.

Layer/Block Mapping in Project Manager

Project - Each import CAD file is contained entirely in a new AGi32 project. The import project name consists of the DWG/DXF filename and a date/time stamp.

Drawing Entities - Each layer / block becomes a drawing entity within the Drawing Entity Type classification of the import project. Color, LineWeight, and LineType are determined by their corresponding entity properties.

Objects - Objects are located within the Object Type classification of the import project by Object Label. Object Labels are determined by layer / block and the number of objects on that layer / block. Format: LayerName_1, LayerName_2, ... , LayerName_N, BlockName_1, BlockName_2, ... , BlockName_N, etc.

Layer/block names are also preserved in the Component Label of each object.

Wireframe Color is determined by the entity color or by layer color (if the entity color is "ByLayer" in Project Manager).

3D Entities

All 3D entities imported into AGi32 are assigned surface properties such as Color/Reflectance or Texture attributes upon import. By default, AGi32 converts all 3D entities to double-sided Planar Objects with 50% grayscale reflectance. The surface types and attributes may be modified during the import process with import mapping (see below), or after import using the Surface Edit command.

Grouping Single Objects

As discussed above, in the Layers and Blocks section, 3D entities may be grouped into a single object in the 3D Import Options section of the Import CAD File dialog. This method of grouping the layers together is only appropriate for entity import (e.g., imported file represents a single complex entity such as a plant, person, furniture, etc.).

When a whole 3D environment is imported in AGi32, grouping 3D Entities into a single object is strongly discouraged. Complex environments grouped together can cause coplanar surfaces (such as end panels within exterior walls) which may cause calculation errors.

AGi32 will import 3D entities composed of the following entity types:

- 3D Face
- Polyface Mesh
- Polygon Mesh
- Regions (holes within are supported)
- 3D Solids (Planar surfaces only), holes within are supported

More information about the limitations and options within each entity type is provided in the Imported Entity Types section.

When 3D entities are grouped into a single object, the wireframe color of all surfaces in the object is converted to Black.

Surface Checks

By default, when AGi32 imports 3D entities as objects, they are inspected for consistent surface orientation and duplicate surfaces. These checks are performed within each object by processing the vertices of a specified percentage of surfaces together within each object. By default, AGi32 uses a value of 0.25 (25%) as its grouping value. This value, the '3D Entity Check Range' may be changed in the 3D Import Options section of the Import CAD File dialog.

- This process is based on the assumption that adjacent surfaces in the model should be within a certain percentage of each other in the CAD file. This may not always be true, in which case the percentage of surfaces checked should be raised by increasing the '3D Entity Check Range'. A 3D Entity Check Range equal to 1.00 will ensure consistent surface orientation and/or no duplicate surfaces within each imported object. However, the larger this number, the longer AGi32 will require to perform these inspections and modifications.
- Files with imported surfaces that are of inconsistent surface orientation or have duplicate surfaces may not render with Adaptive Subdivision enabled.

Object naming convention

Object Labels are determined by their layer / block name and the number of objects on that layer / block.

Format: LayerName_1, LayerName_2, ... , LayerName_N, BlockName_1, BlockName_2, ... , BlockName_N, etc. (e.g. 2 objects on Layer Skin would be named Skin_1 and Skin_2).

Layer/block names are also preserved in the Component Label of each object. For example, in a group labeled ImportCar, layer Wheels would become the name of the Component containing the wheel surfaces, layer Body would become the name of the Component containing the body surfaces, etc.

Using 3D Entity Models as Library Objects

Composite objects can be saved, to use and reuse, as library objects. Once you save an imported object as a library object, you can reuse it in any job file, complete with its associated material properties.

Note: It is strongly recommended you create your library objects at a true scale, so that they do not need to be scaled repeatedly with each reuse. We recommend checking the scale of your imported object quickly, in a test file, before you spend a lot of time assigning the desired surface materials.

Import Mapping (3D Entities)

Entities that are imported as AGi32 objects are assigned surface properties during import. The user can customize these properties by applying an import map. Otherwise, a default import map is used.

Import Maps may be saved and opened. The folder in which they are saved can be set in System Settings. By default, it is located in My Documents\AGi32 \CADFiles_Import_Maps. Additionally, an import map is automatically saved for each imported file. If the same file is imported again, the previously saved import map is automatically loaded. Import maps may be deleted by deleting the import map files directly in My Computer.

Import Filters

- An import map is composed of one or more import filters.
- An import filter is used to apply AGi32 surface properties (such as surface type, reflectance, etc.) to entities with specified CAD properties (color, layer,

etc.). Import filters can be thought of as translators that convert CAD properties into AGi32 surface properties. Object properties (component label, wireframe color) cannot be specified by import filters.

- An import filter is applied to each (every) entity and all its sub-entities.
- Only one filter may be applied to each entity (each filter applies an entire set of surface properties).
- If an entity meets the criteria for more than one filter, the entity applies first filter it finds (filters order matters).
- If an entity does not meet the criteria of any filter, the default filter is applied.

Surface Properties (see Surface Edit command for detailed information)

Surface Property	Comment
Surface Label	Optional surface labels provide for naming objects with consistent name, allowing for easy identification in Project Manager
Removed	Selecting this option removes selected surfaces from calculation consideration (surfaces become invisible)
Surface Type	Available Surface Types: Double Sided, Single Sided, Daylight Transition Glass (Transparent and Diffuse), Daylight Transition Opening, Roadway Pavement Single Sided (Direct Flux Only) and Roadway Contributor Single Sided)
Daylight Exterior	Selecting this option assigns selected surfaces as Daylight Exterior surfaces (impacts daylighting computations only).
Color / Reflectance	Specify a Color / Reflectance value for the selected surfaces. Clicking in the Texture cell results in the Color / Reflectance Selector dialog being displayed.
Texture	Specify a Texture image to be applied to the selected surfaces. Clicking in the Texture cell will reveal the Select texture dialog. An preview thumbnail of the texture is displayed once a texture image is selected.
Transmittance	Double Sided surfaces may be assigned a diffuse transmittance value (e.g. they behave like frosted glass).
Color / Transparency	When the Glass surface type is specified, you can specify a Color / Transparency value for the selected surfaces. Clicking in either cell results in the Color / Transparency dialog being displayed.
Specularity	Specularity values range between zero and 1, where any value greater than zero results in a high gloss surface in Ray Trace imagery.
Glossiness	Glossiness values range between zero and 1, where any value greater than zero in a low gloss finish in Ray Trace imagery (must be assigned in conjunction with specularity).
Color Bleed	Color Bleed values range between zero and 1, where any value less than one results in lesser color reflection (used to mitigate saturated color reflection between surfaces for more realistic appearing renderings).
Mesh Level	Mesh level changes results in finer or coarser surface sampling (e.g. change from default settings).
Color / Luminance	Specify a Luminance value (in Cd/M^2) for self emitting surfaces. By default, luminance color equals surface color but they may be different. Clicking in either cell results in the Color / Transparency dialog being displayed.
Direct Flux Only	Selecting this option assigns selected surfaces as Direct Flux Only surfaces (surfaces receive no reflected light); useful for optimization purposes.

Default Map/Filter

- Apply to ALL entities ....
- Surface Type : Double-Sided
- Reflectance : 0.5
- Color : RGB - 128 128 128 (middle gray)

Filter Types

- Apply to ALL entities ....
- If Entity has Color ....
- If Entity has LineType ....
- If Entity has LineWeight ....

- If Entity is on Layer ....
- If Entity is within Block ....
- If Entity is on Layer where the Layer Name contains ....
- If Entity is within Block where the Block Name contains ....

Import Mapping tabs

There are two ways to create import maps: by layer and advanced.

By layer import mapping allows the user to quickly apply surface properties to entities based on the entities' layer (automatically creating layer filters). The CAD Viewer shows a preview of the currently selected layers. To quickly select and clear layers use the Select All and Clear Selected buttons. Alternately you may use the Ctrl-A keystroke combination to toggle the selection between Select All and Clear All.

Advanced import mapping provides more powerful control by providing access to all available filter types. However, the CAD Preview is not updated when creating advanced import maps.

CAD Viewer

CAD Viewer Usage

- The CAD Viewer allows users to preview a drawing before importing.
- The CAD Viewer responds to layer selection while selecting layers to import as well as selecting layers for import mapping.
- The CAD Viewer is resizable and can overlap the Import CAD File dialog.
- The CAD Viewer displays the view the CAD file was saved under, including any shading options that may have been applied in the CAD software. At this time, it is not possible to rotate the 3D view to any other WCS display.

CAD Viewer Functions

	Refresh	This button refreshes the view to reflect the current layer selection.
	AutoRefresh	This button allows the CAD Viewer to refresh automatically after each layer selection. AutoRefresh is the default setting. **Note:** This may decrease performance with large or complicated CAD files.
	Zoom Extents	Zooms to the extents of the currently selected layers.
	Zoom In	Zooms in (zoom factor =2)
	Zoom Out	Zooms out (zoom factor =0.5)
	Show	The Show menu displays whether to display 3D and/or 3D entities in the CAD Viewer. This functionality allows you to visually confirm whether entities will import as 2D drawing entities or 3D objects.
	Show 2D Entities	Show 2D Entities is applied by default. This setting determines whether the CAD viewer displays 2D entities.
	Show 3D Entities	Show 3D Entities is applied by default. This setting determines whether the CAD viewer displays 3D entities.

Color Mapping

AutoCAD Color Indexes (ACI) are automatically converted to Windows color numbers. Please note, AutoCAD only uses 256 colors in its ACI table. Windows uses either; 16, 256, 16 bit or 24 Bit colors. Consequently, there may be slight variation in the color between AutoCAD ACI colors and AGi32 colors. As of AutoDesk software versions 2004 and newer, AutoDesk products may have wireframe colors specified using any 24 bit RGB value. AGi32 will be able to match these colors exactly when the file is imported in AGi32.

Line Type Mapping

When importing AutoCAD created files into AGi32, AutoCAD line types must be mapped to one of AGi32's five available line types. The table below shows details of mapping AutoCAD's standard line types into AGi32.

AGi32 Line Type	ACAD Line Type
SOLID	CONTINUOUS
DASH	CENTER CENTER2 CENTERX2 DASHED

	DASHED2 HIDDEN HIDDEN2 HIDDENX2 PHANTOM PHANTOM2 PHANTOMX2
DOT	DOT DOT2 DOTX2
DASHDOT	BORDER BORDER2 BORDERX2 DASHDOT DASHDOT2 DASHDOTX2
DASHDOTDOT	DIVIDE DIVIDE2 DIVIDEX2

All other linetypes are imported as Solid linetypes in AGi32.

Font Mapping

AutoCAD uses its own fonts and there is no correlation with Windows fonts. Since AGi32 uses Windows fonts, imported text is converted to the current AGi32 default system font. This default font setting is located in the Tools-System Settings dialog and can be changed as desired. Imported text can be changed using the Modify-Drawing-Edit Text command.

Note: If the AutoDesk font described in the imported CAD file is found on the end users computer (in the Windows Font folder), AGi32 will use the CAD font instead of the AGi32 default system font.

XREF Drawings

External References (XREFs) found in the imported file will automatically be imported if the XREF file can be found.

- If the associated XREF file(s) can not be found, the program will display a message and allow you to Browse for the XREF. This may occur if the CAD file and/or XREF file has been moved from its original location. If the XREF can not be found via browsing, you can elect to Ignore the XREF or Cancel the import process.
- XREF's behave like blocks and, therefore, are affected by the block settings and hierarchy including the 'Keep High/Low Level Blocks' options (see Blocks and XREFs above).
- 'Clipping' is not supported in XREF's in AGi32. CAD allows you to 'clip' an XREF with a polygonal boundary so only a portion of the XREF is visible (think viewport). AGi32 imports all entities in the XREF regardless of what is visible in CAD (i.e. clipping is ignored).
- Another option to include XREFs in the import process is to bind the XREF in CAD.

Binding XREFs

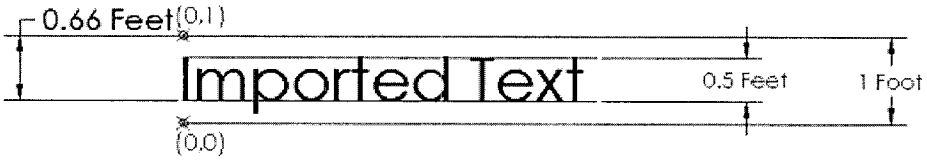
1. In AutoCAD, the command BIND is used to bind an external reference (this can be done in the XREF dialog in CAD).
2. Binding permanently inserts the XREF as a block into the original drawing. Always do the bind on a copy of the original drawing (so that you do not lose your "live" links).
3. You cannot BIND in XREFs with proxy objects or invalid names (names with [ ] # or @ characters.

Proxy Entities

Proxy Entities found in AutoCAD are not imported into AGi32. Proxy Entities, by definition, are temporary placeholders in AutoCAD that stand in for certain entities created in Autodesk Desktop software, such as Architectural Desktop or Mechanical Desktop.

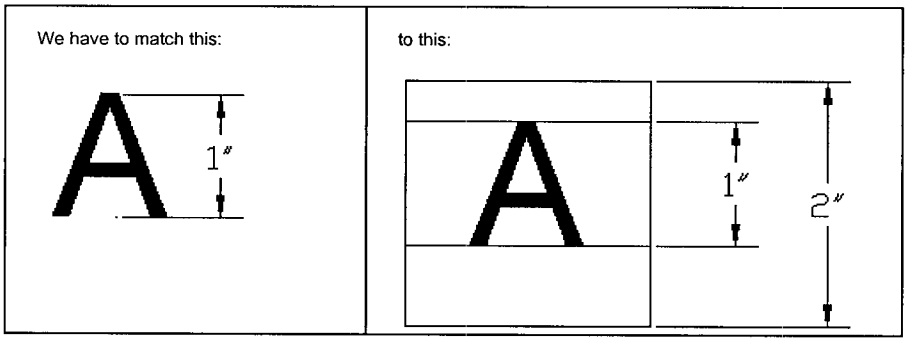
Proxy graphics may show up in the imported drawing when viewed in AutoCAD and CAD Viewer but will not be imported in AGi32.

Text Size and Offset Factors



Text Size and Offset Factor are applied to text in DXF files when they are imported into and exported from AGi32. These parameters are applied to the text so that in AGi32 it looks similar to its original appearance in CAD.

The text size in AGi32 determines the size of the character block containing the font glyph and top and bottom leading spaces. In CAD software, the text is created without any top and bottom leading spaces.



Text Size Factor adjusts the CAD Text Size to correspond to the AGi32 Text Size, which includes the top and bottom leading spaces. An example illustrates these concepts.

- The default Text Size Factor is 2
- AGi32 Text Size = CAD Text Size * Text Size Factor
- AGi32 Text Size = 0.5 * 2 = 1 (used above)

Offset Factor offsets the text to account for the bottom leading space in AGi32.

- The default Offset Factor is 1.5 (1/0.66).
- YNEW = Y – (AGi32 Text Size / Offset Factor)
- YNEW = 1 – (1/1.5) = 0.33